ACCESS 2002

ADVANCED

ADVANCED

SECTION 1 -
USING ACTIVEX CONTROLS

In this section, you will learn how to:

- Work with ActiveX controls
- Register an ActiveX control
- Add an ActiveX control
- Attach an ActiveX control

WORKING WITH ACTIVEX CONTROLS

✎ Discussion

In addition to the standard controls, such as option groups, combo boxes, and list boxes, you can add ActiveX controls to a form or report. ActiveX controls are small applications that can improve the appearance or functionality of a form or report. A commonly used ActiveX control is the calendar control. This control appears as a calendar in a form or report. The calendar control can be bound to a **Date** field in the form. Instead of having to type in the dates, you can just click the desired date on the calendar.

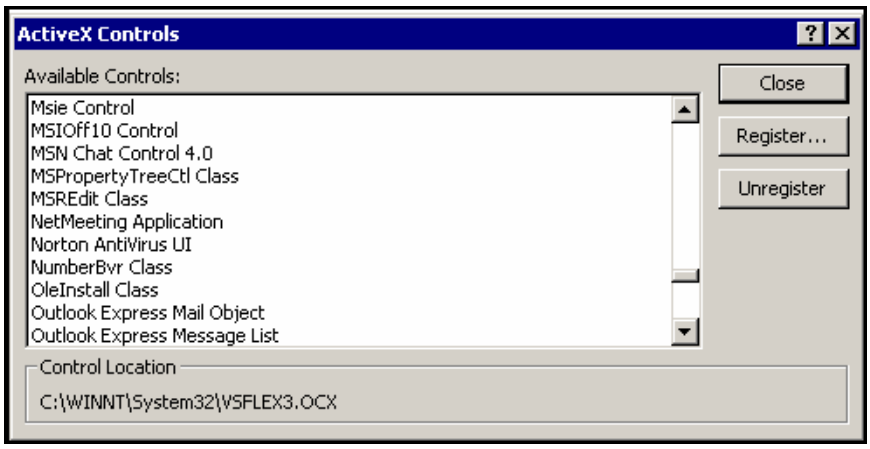
●	ActiveX controls were referred to as Custom Controls in previous versions of Access.

●	The calendar control is included with your Access application and is available as long as you selected it during the Access Setup routine when you installed the software. Other ActiveX controls are available from the Access Developer's Edition as well as from vendors other than Microsoft. ActiveX controls have an **.ocx** file extension.

REGISTERING AN ACTIVEX CONTROL

✎ Discussion

Before you can use an ActiveX control, it must first be registered with Access. Once it is registered, the name of the control appears in the ActiveX Controls dialog box. Access then knows that the control exists and where it is located on the computer or network.



The ActiveX Controls dialog box

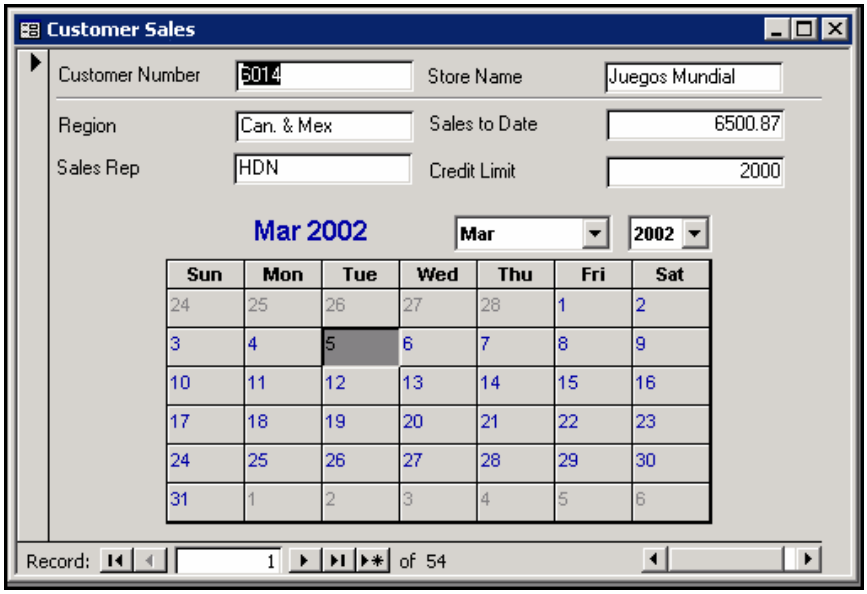
➡ Procedures

1. Open the desired database.
2. Select the **Tools** menu.
3. Select the **ActiveX Controls** command.
4. Select **Register**.
5. Select the **Look in** list.
6. Select the drive where the control is located.
7. Select the folder where the control is located.
8. Select the file containing the desired ActiveX control.
9. Select **OK**.
10. Select **Close**.

ADDING AN ACTIVEX CONTROL

✎ Discussion

After you register an ActiveX control with Access, you can add it to a form or report. Once it is added, the ActiveX control can be used like any other control. You can move and size it, as necessary, to fit on the form or report. You can also display and change the properties of an ActiveX control.



The ActiveX calendar control

- You cannot add an ActiveX control to a form if it has not been registered with Access.

➡ Procedures

1. Open the desired database.
2. Select the **Insert** menu.
3. Select the **ActiveX Control** command.
4. Select the desired ActiveX control.
5. Select **OK**.

ATTACHING AN ACTIVEX CONTROL

✎ Discussion

Depending on the ActiveX control and its function, you can attach it to an existing table in Access. For example, you can attach the calendar control from Microsoft Windows 95 to a **Date** field in Access. The calendar will then display the date on a calendar page for the appropriate month and year.

➡ Procedures

1. Open the desired database.
2. Open the desired form or report in **Design** view.
3. Select the ActiveX control.
4. Click the **Properties** button  on the **Form Design** toolbar.
5. Select the **Data** tab.
6. Select the **Control Source** list.
7. Select the desired data source.
8. Click the **Close** button to close the property sheet.

SECTION 2 -
WORKING WITH INDEXES

In this section, you will learn how to:

- View indexes
- Create a single field index
- Create a multiple field index
- Delete an index
- Create a multiple field primary key

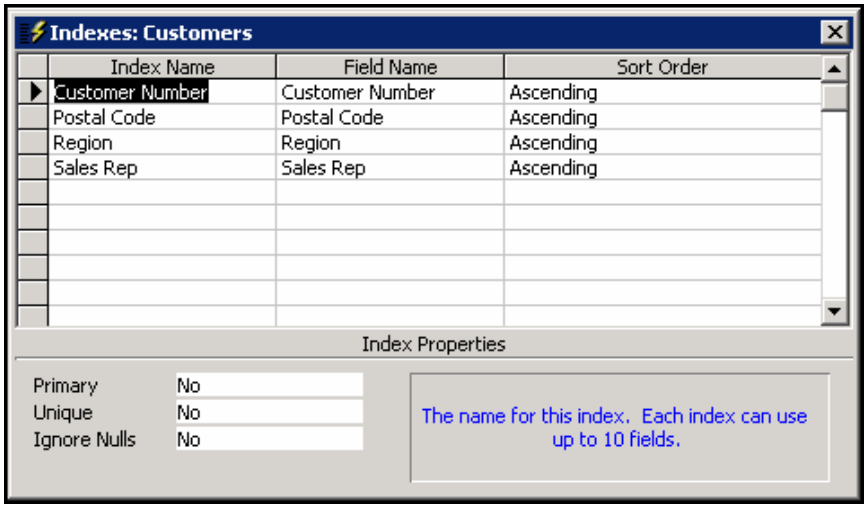
VIEWING INDEXES

Discussion

You can use indexes to help Access perform quicker sorts and searches. You should index the fields you use regularly to sort or search for data.

Indexes have important properties that are listed in the following table:

Property	Description
Primary	Designates whether or not the index is the primary key for the table. In other words, this property designates whether or not the data in the indexed field will be used to uniquely identify each record. If this property is set to Yes, the **Unique** property must be set to Yes.
Unique	Designates whether or not the values in the indexed field must be unique for each record. You can set this property to Yes without setting the **Primary** property to Yes.
Ignore Nulls	Determines whether or not records with null values in the indexed field are included in the index. The default setting is No, which means that null values are included.



The Indexes window

●	You can also display the Indexes dialog box by selecting the **View** menu and then selecting the **Indexes** command when a table is open in **Design** view.

●	You cannot create an index for **Memo**, **OLE,** or **Hyperlink** fields.

●	If there are a limited number of different values in a field, then indexing the field does not noticeably speed up searches and sorts. In fact, it may slow down data entry and editing.

➡ Procedures

1. Open the desired database.
2. Open the table in **Design** view.
3. Click the **Indexes** button  on the **Table Design** toolbar.

CREATING A SINGLE FIELD INDEX

✎ Discussion

You should index at least one field for each table to maximize performance. You should select a field you are likely to use in searches and sorts.

When you change the **Indexed** property of a field to Yes, you can choose whether or not to allow duplicates. If you choose the **No Duplicates** option, Access will not allow duplicate values to be entered into the field.

➡ Procedures

1. Open the desired database.
2. Open the desired table in **Design** view.
3. Select the field you want to index.
4. Select the **Indexed** property.

5. Select the **Indexed** list.
6. Select the desired indexing option.

CREATING A MULTIPLE FIELD INDEX

✎ Discussion

If you often search on a combination of fields, you can create an index on multiple fields to speed up the searches and sorts. For example, if you want to sort customers in order by region and alphabetically within each region, you can create an index on the **Region** and **Customer** fields.

You must create a multiple field index in the Indexes window. All the fields you want to include in the index should be listed under the same index name. You can include up to ten fields in an index.

➥ Procedures

1. Open the desired database.
2. Open the table in **Design** view.
3. Click the **Indexes** button  on the **Table Design** toolbar.
4. Select the first blank field in the **Index Name** column.
5. Type the name of the index.
6. Press **[Tab]**.
7. Select the **Field Name** list.
8. Select the field you want to index.
9. Press **[Tab]**.
10. Select the **Sort Order** list.
11. Select the desired sort order.
12. Select the next blank field in the **Field Name** column.
13. Select the **Field Name** list.
14. Select the field you want to index.
15. Press **[Tab]**.

16. Select the **Sort Order** list.
17. Select the desired sort order.
18. Close the Indexes window.

DELETING AN INDEX

✎ Discussion

Indexes take up disk space and can slow down the process of adding, editing, and deleting records. In fact, if there are a limited number of different values in a field, the sort or search speed is not significantly increased by indexing the field. Therefore, you can delete any unwanted indexes to increase efficiency.

➥ Procedures

1. Open the desired database.
2. Open the table in **Design** view.
3. Click the **Indexes** button  on the **Table Design** toolbar.
4. Point to the row selector to the left of the row containing the index you want to delete.
5. Click the row selector to the left of the row containing the index you want to delete.
6. Press **[Delete]**.

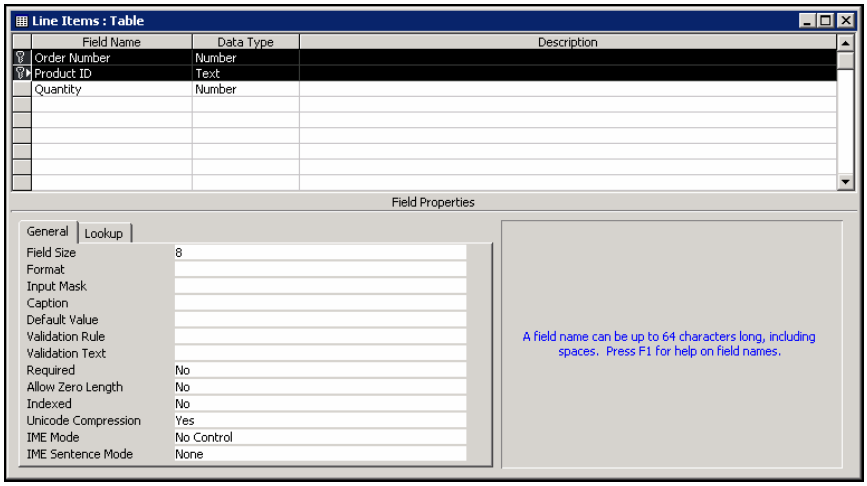
CREATING A MULTIPLE FIELD PRIMARY KEY

✎ Discussion

There are times when it is not possible to uniquely identify fields with the values in a single field. Therefore, Access allows you to create a primary key using multiple fields.

When you set a primary key using multiple fields, the combination of the information in the fields must be unique. For example, you have a table that contains line items for customer orders. The order number is entered for each customer. Because you will

most likely have multiple customers with multiple orders, the order number associated with each customer may not be unique. Also, a particular item number may appear in multiple orders. Therefore, the item numbers may not be unique. Thus, you cannot create a primary index on either of these fields by themselves. The solution is to create a primary key index on both fields, which allows each item to appear only once on each order.



Creating a multiple field primary key

➥ Procedures

1. Open the desired database.
2. Open the table in **Design** view.
3. Point to the row selector to the left of the row containing the first field for the primary key.
4. Click the row selector to the left of the row containing the first field for the primary key.
5. Hold the **[Ctrl]** key and point to the row selector for the second primary key field.
6. Click the row selector for the second primary key field and release the **[Ctrl]** key.
7. Click the **Primary Key** button  on the **Table Design** toolbar.

SECTION 3 -
REPLICATING DATABASES

In this section, you will learn how to:

- Replicate a database
- Synchronize database replicas
- Resolve replication conflicts

REPLICATING A DATABASE

✎ Discussion

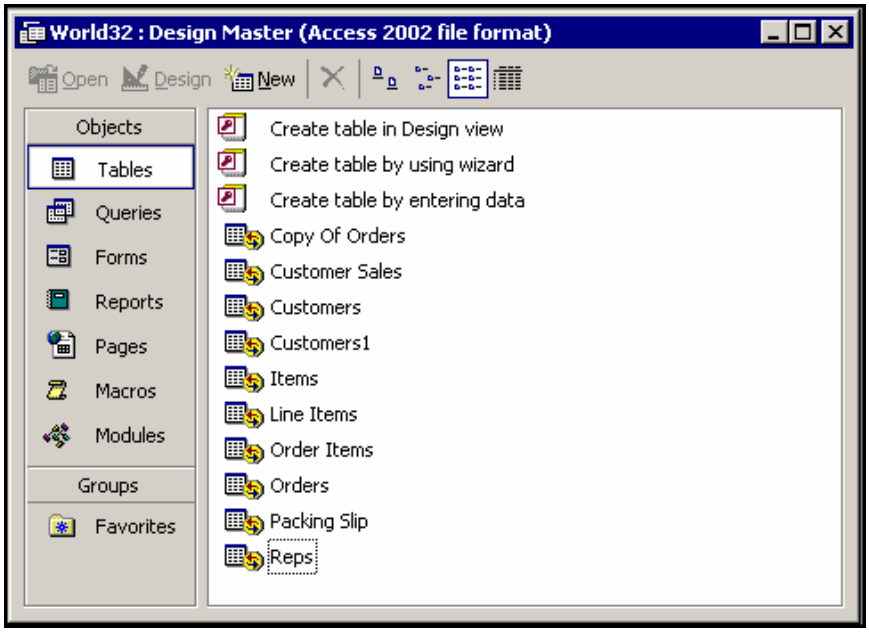
Replicating a database allows you to share a database. It is useful to share a database across a network or create copies for remote users, such as a mobile sales force. While you could copy the file, there is an advantage to replication. Any changes to data made in a replica can be updated in the original database and other replicas through synchronization. Additionally, any design changes to the original database are distributed among the replicas.

You can create a single replica or multiple replicas. You may want to create a single replica if you want to place the database on a remote computer, such as a laptop, or if you want to create a backup of the database. You can create multiple replicas to distribute the database among members of a department or a workgroup. The databases can then be stored on the hard drives of the machines, which decreases activity on a network. All the replicas made from a single Design Master make up a replica set.

When you replicate a database, the original database becomes the Design Master. The Design Master and each of the replicas function as standalone databases. Each user can add, edit, or delete data in their replica. These changes in the replicas can then be synchronized with the Design Master and other replicas in the replica set. Each user can add, edit, or delete objects in the database, such as tables, queries, reports, or forms. However, these changes are not copied to the Design Master during synchronization. Only the structural changes made in the Design Master are distributed to the replica set.

When a database becomes a Design Master, Access adds hidden fields and tables to the database to facilitate synchronization. These hidden fields increase the size of the database. Certain properties are available on the property sheets of table and query database objects as well. The **Replicable** property applies to a table or query object and indicates whether the **Replicable** property object can be replicated. You can keep table and/or query objects in the Design Master from being replicated by deselecting the **Replicable** property. This option can be useful if there are tables in the Design Master that you do not want to include in the replicas, such as those containing sensitive information, or information that would be of no use to the replica user and would only take up disk space. In this case, you would be creating a partial replica.

When you create a replica for the first time, the current database is converted to a Design Master. After the first time, you simply create another replica.



The Design Master database window

- During the replication procedure, you can create a backup of the original file. This option is useful because you will always have a second copy of your database in case the original is lost, damaged, or becomes corrupted.

- The Microsoft Briefcase Replication is not included in the default installation. You can choose to install it with the initial installation, or it can be installed by itself at a later time.

- If you have **AutoNumber** fields in a table, their behavior changes in database replicas. While the existing numbers do not change, any new records added generate random numbers. If the incremental numbering in the **AutoNumber** field is crucial, use another **Date/Time** field to produce incremental numbers.

- If you plan to convert an Access 2002 database to an Access 97 database, you must perform the conversion before it becomes a part of a replica set. To convert an Access 2002 database to an Access 97 database, select the **Tools** menu, point to the **Database Utilities** command, point to the **Convert to Database** command, and then select the **To Prior Access Database Version** command.

➡ Procedures

1. Open the desired database.
2. Select the **Tools** menu.
3. Point to the **Replication** command.
4. Select the **Create Replica** command.
5. Select **Yes**.
6. Select **Yes**.
7. Type a name for the replica in the **File name** text box.
8. Select the **Save in** list.
9. Select the drive where you want to save the replica.
10. Select the folder where you want to save the replica.
11. Select **OK**.
12. Select **OK**.

SYNCHRONIZING DATABASE REPLICAS

✎ Discussion

When changes have been made to replicated files or the Design Master, they can then be synchronized. You can synchronize a replica with the Design Master or another replica in the replica set.

Synchronization is bi-directional, meaning that all changes are exchanged between the files in both directions.

➡ Procedures

1. Open the Design Master database window.
2. Select the **Tools** menu.
3. Point to the **Replication** command.
4. Select the **Synchronize Now** command.
5. Select the **Directly with Replica** list.

6. Select the desired replica file.
7. Select **OK** to begin the synchronization process.
8. Select **Yes** to complete the synchronization process.
9. Select **OK**.

RESOLVING REPLICATION CONFLICTS

✎ Discussion

When replicas are synchronized with one another, replication conflicts can occur. Conflicts are usually generated when the same data in multiple files has been changed in different ways. If a table has replication conflicts, Access opens a message box when the file is opened. You can choose to resolve the conflicts at this point or resolve them later using the **Resolve Conflicts** feature.

Access stores replication conflicts in a conflicts table in the database. This table holds all records with conflicts. When you resolve conflicts, Access displays the existing record and the conflicting record next to each other. If you choose to keep the existing record, Access overwrites the conflicting record. If you choose to keep the conflicting record, Access overwrites the existing record. Either way, you cannot undo the change.

➡ Procedures

1. Open the Design Master database window.
2. Select the **Tools** menu.
3. Point to the **Replication** command.
4. Select the **Resolve Conflicts** command.
5. Select **Resolve Conflicts**.
6. Select **Keep Existing Record** or **Overwrite with Conflict Record**.
7. Select **Yes** or **No** to confirm.
8. Repeat steps 5-7 until all conflicts are resolved.
9. Select **OK**.
10. Select **Close** to close the Resolve Replication Conflicts dialog box.

SECTION 4 -
CREATING MACROS

In this section, you will learn how to:

- Work with macros
- Open the Macro Design window
- Create a macro
- Assign an argument to an action
- Save a macro
- Use single step mode for testing
- Run a macro
- Edit an existing macro
- Run a macro using the Tools menu
- Create a macro using the Macro Builder

WORKING WITH MACROS

✎ Discussion

A macro performs a set of commands in sequence. While macros in word processors and spreadsheets are used mainly to duplicate keystrokes or mouse movements, macros in Access often automate an action or a series of actions. Such actions include opening tables, printing forms, finding records, or applying filters. Macros can even be used to add command buttons, create menus and toolbars, and build complete applications.

Macro commands in Access consist of an action and its arguments. The action is the task to be performed, such as opening a form. The arguments determine the specifics for the action, such as which form to open.

You create a macro in Access from the Macro Design window. This window has two panes. The upper pane, called the **Action** pane, contains a design grid. The design grid can contain up to four columns. The **Action** and **Comment** columns are always displayed. The **Action** column contains one of the many available macro commands. In the **Comment** column, you can type a description of the action. Comments are helpful when you are editing macros that contain many actions. You can also display the **Macro Name** and **Condition** columns. The **Macro Name** column contains a name for a macro that can be referred to during events, such as clicking a command button. You can execute a macro conditionally by adding a statement to the **Condition** column. The lower pane, called the **Action Arguments** pane, contains the arguments. The arguments listed change, depending on what action is selected.

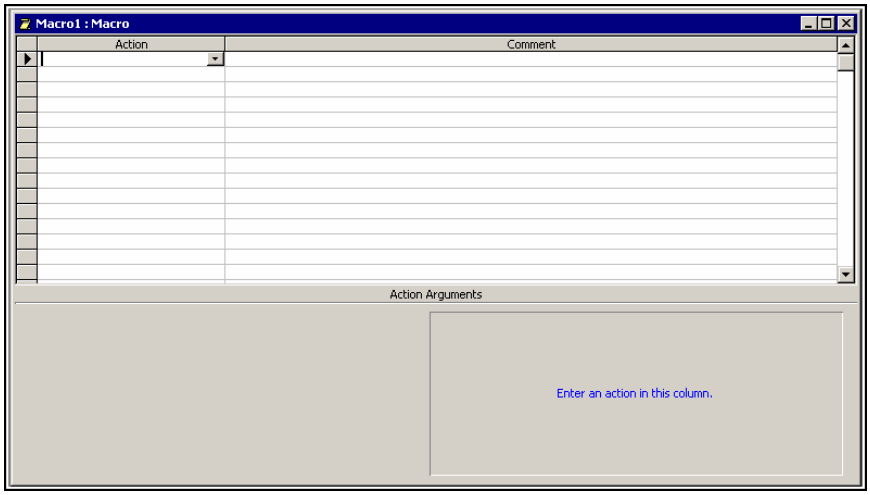
- The right side of the **Action Arguments** pane displays helpful information about the selected action or action argument.

OPENING THE MACRO DESIGN WINDOW

✎ Discussion

The design grid in the **Action** pane of the Macro Design window contains at least two columns: **Action** and **Comment**. In the **Action** column, you enter the actions you want the macro to perform. You can either type the name of the action or select one from the list. In the **Comment** column, you describe the purpose of the action. Adding a comment may not seem important when a macro is first created, but it can prove helpful if you need to edit the macro at a later time.

The **Action Arguments** pane appears at the bottom of the window and varies according to the action selected in the **Action** pane. Most actions require one or more arguments. In the **Action Arguments** pane, the arguments pertaining to an action appear. The arguments detail exactly what function the action performs.



The Macro Design window

➡ Procedures

1. Open the desired database.
2. Display the **Macros** object list.
3. Select the **New** button on the Database window toolbar.

CREATING A MACRO

✎ Discussion

When you select a field in the **Action** column, a list of actions appears. Most of the actions are self-explanatory and have equivalent menu commands. For instance, the **OpenQuery** action opens a query in **Datasheet** or **Design** view, depending on how the arguments are set.

Other actions can be performed only in macros or in more complex programming modules. The **AddMenu** action, for example, creates a custom menu to appear on a custom menu bar. This action is not available from the menu commands.

A macro can include up to 999 actions. You place each action in a separate row in the design grid in the order in which they are to be performed. For example, a **Maximize** command maximizes the window opened in the step immediately preceding it.

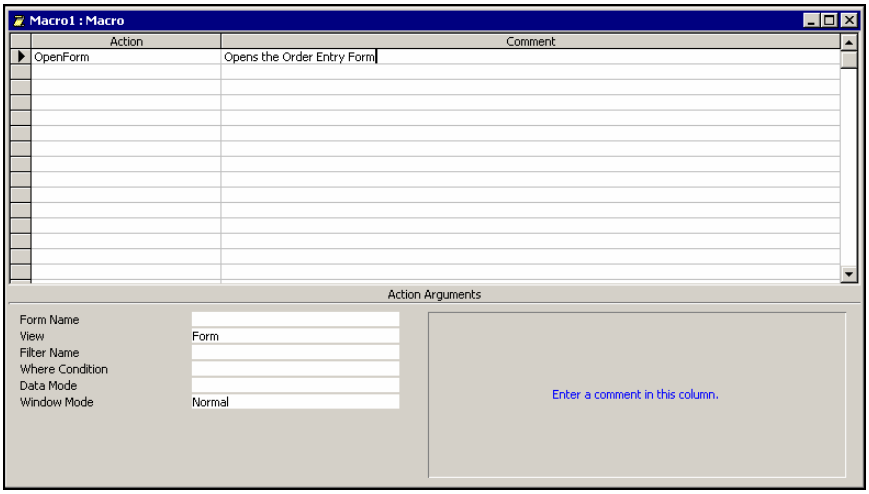
You use the **Comment** column to enter a description of the action to be performed. This field can contain up to 255 characters. While comments are optional, it is beneficial to enter a description of the action. This field is helpful if you want to modify the macro at a later time.

The following table provides a brief description of each action:

Action	Description
AddMenu	Adds a drop-down menu item to a custom menu bar.
ApplyFilter	Applies a filter or query to restrict or sort records.
Beep	Sounds a tone through the computer's speaker.
CancelEvent	Cancels the action that initiated the macro.
Close	Closes a specified window.
CopyObject	Copies the specified object to a different database or the same database under a new name.
DeleteObject	Deletes the specified object.
Echo	Specifies if the screen is to be updated while the macro is running.
FindNext	Locates the next record that meets criteria specified by a find.
FindRecord	Locates the first record following the current record that meets criteria specified in the arguments.
GoToControl	Activates the specified field or control.
GoToPage	Activates the first control on a specified page.
GoToRecord	Makes the specified record the current record.
Hourglass	Changes the mouse pointer to an hourglass while the macro is running.
Maximize	Enlarges the active window to fill the screen.
Minimize	Reduces the active window to a button on the taskbar.
MoveSize	Moves and/or resizes the active window.
MsgBox	Opens a message box containing a warning or informative text.
OpenDataAccessPage	Opens a data access page in the selected view.
OpenDiagram	Opens a project diagram.

Action	Description
OpenForm	Opens a form in the selected view.
OpenModule	Opens a module at the selected procedure.
OpenQuery	Opens a query in the selected view.
OpenReport	Opens a report in the selected view.
OpenStoredProcedure	Opens a project stored procedure in the selected view.
OpenTable	Opens a table in the selected view.
OpenView	Opens a view in the current project.
OutputTo	Outputs data in the specified object to Excel text (.XLS), rich-text (.RTF), or MS-DOS text (.TXT).
PrintOut	Prints the specified datasheet, form, report, or module.
Quit	Quits Access.
Rename	Renames the specified database.
RepaintObject	Completes any pending screen updates for the active or specified object.
Requery	Updates the data in a specified control of an active object or updates itself if no control is specified.
Restore	Restores a maximized or minimized window to its previous size.
RunApp	Runs an application from within Access.
RunCode	Runs a Microsoft Access Basic Function procedure.
RunCommand	Performs a menu command.
RunMacro	Runs a macro.
RunSQL	Runs an action query using the corresponding SQL statement.
Save	Saves the specified object. If an object is not specified, the active object is saved.
SelectObject	Selects an object.
SendKeys	Sends keystrokes directly to Access or to an active Windows application.
SendObject	Includes a specified object in an electronic mail message to be viewed and forwarded.
SetMenuItem	Sets the appearance of a command that appears in a custom menu.
SetValue	Sets the value of a field, control, or property on a form or report.

Action	Description
SetWarnings	Turns system messages on or off.
ShowAllRecords	Removes any applied filter from the applicable active object and displays all records.
ShowToolbar	Displays or hides a toolbar.
StopAllMacros	Stops all currently running macros.
StopMacro	Stops the currently running macro.
TransferDatabase	Imports or exports data between an Access database and another database.
TransferSpreadsheet	Imports or exports data between the active database and a spreadsheet file.
TransferText	Imports or exports text between the active database and a text file.



Adding an action and a comment to a macro

- You can use the **[Tab]** key and the **[Shift+Tab]** key combination to navigate between the **Action** and **Comment** columns.

➡ Procedures

1. Open the desired database.
2. Open the Macro Design window.

3. Select the first blank field in the **Action** column.
4. Select the **Action** list.
5. Select the desired action.
6. Select the corresponding field in the **Comment** column next to the action.
7. Type a description for the action.

ASSIGNING AN ARGUMENT TO AN ACTION

✎ Discussion

Most action arguments have a default list of available arguments. For example, the **View** argument field for the **OpenReport** action contains a list with the **Print**, **Design,** and **Print Preview** arguments. For action arguments without a list, you can type the argument into the argument field. You can enter up to 255 characters into the argument field.

Some arguments are required. For example, you must select the **Form Name** argument for the **OpenForm** action. Other arguments, such as **Filter Name**, are not required for the **OpenForm** action. If a required argument is missing, the macro stops when the action containing the missing argument is encountered.

In some cases, a default argument is used. For example, the **View** argument defaults to **Form** for the **OpenForm** action. Other arguments are ignored if they are not selected. For example, the **Filter Name** argument allows you to select a query to apply to the form as a filter. If you do not enter a query name, all the records appear.

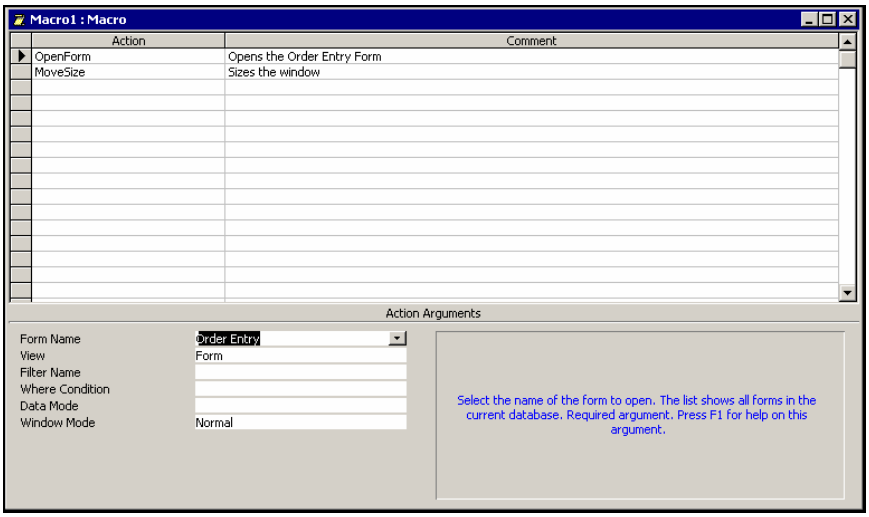
With forms, you can also specify a **Where Condition** argument. This argument acts as a filter, without using an actual query. For example, the argument **[Credit Limit] = 1000** limits records to those with a credit limit of $1000. You can type the condition directly into the argument box or click the **Build** button in the argument box to open the Expression Builder dialog box.

Access displays a helpful message explaining the selected argument to the right of the argument boxes. If you press the **[F1]** key while the insertion point is in the **Action Arguments** pane of the Macro Design window, a help window for the argument opens with additional information.

The available macro actions and their associated arguments are listed in the following table. When the text **(req.)** appears next to an item in the **Arguments** column, either an argument or a selection from a list is required for the macro to run properly:

Action	Arguments
AddMenu	Menu Name (req.), Menu Macro Name (req.), Status Bar Text
ApplyFilter	Filter Name, Where Condition (one or both req.)
Beep	None
CancelEvent	None
Close	Object Type, Object Name, Save (req.)
CopyObject	Designation Database, New Name, Source Object Type, Source Object Name
DeleteObject	Object Type, Object Name
Echo	Echo On (req.), Status Bar Text
FindNext	None
FindRecord	Find What (req.), Match (req.), Match Case (req.), Search (req.), Search As Formatted (req.), Only Current Field (req.), Find First (req.)
GoToControl	Control Name (req.)
GoToPage	Page Number (req.), Right, Down (if a Right is specified, a Down must be specified and vice versa)
GoToRecord	Object Type, Object Name, Record (req.), Offset
Hourglass	Hourglass On (req.)
Maximize	None
Minimize	None
MoveSize	Right, Down, Width, Height
MsgBox	Message, Beep (req.), Type (req.), Title
OpenForm	Form Name (req.), View (req.), Filter Name, Where Condition, Data Mode (req.), Window Mode (req.)
OpenModule	Module Name, Procedure Name
OpenQuery	Query Name (req.), View (req.), Data Mode (req.)
OpenReport	Report Name (req.), View (req.), Filter Name, Where Condition
OpenTable	Table Name (req.), View (req.), Data Mode (req.)
OutputTo	Object Type (req.), Object Name, Output Format, Output File, Auto Start, Template File
PrintOut	Print Range (req.), Page From (req.), Page To (req.), Print Quality (req.), Copies (req.), Collate Copies (req.)

Action	Arguments
Quit	Options (req.)
Rename	New Name (req.), Object Type, Old Name
RepaintObject	Object Type, Object Name
Requery	Control Name
Restore	None
RunApp	Command Line (req.)
RunCode	Function Name (req.)
RunCommand	Command (req.)
RunMacro	Macro Name (req.), Repeat Count, Repeat Expression
RunSQL	SQL Statement (req.), Use Transaction (req.)
Save	Object Type, Object Name
SelectObject	Object Type (req.), Object Name (req.), In Database Window (req.)
SendKeys	Keystrokes (req.), Wait (req.)
SendObject	Object Type, Object Name, Output Format (req.), To (req.), Cc, Bcc, Subject, Message Text, Edit Message (req.), Template File
SetMenuItem	Menu Index (req.), Command Index (req.), Subcommand Index, Flag (req.)
SetValue	Item (req.), Expression (req.)
SetWarnings	Warnings On (req.)
ShowAllRecords	None
ShowToolbar	Toolbar Name (req.), Show (req.)
StopAllMacros	None
StopMacro	None
TransferDatabase	Transfer Type (req.), Database Type (req.), Database Name (req.), Object Type (req.), Source (req.), Destination (req.), Structure Only (req.)
TransferSpreadsheet	Transfer Type (req.), Spreadsheet Type (req.), Table Name (req.), File Name (req.), Has Field Names (req.), Range
TransferText	Transfer Type (req.), Specification Name (req.), Table Name (req.), File Name (req.), Has Field Names (req.), HTML Table Name



Assigning an argument to an action

- You can press the **[F6]** key to toggle between the **Action** pane and the **Action Arguments** pane. You can use the **[Tab]** key and the **[Shift+Tab]** key combination to navigate through the argument fields.

➡ Procedures

1. Open the desired database.
2. Open the Macro Design window.
3. Select the action in the **Action** column to which you want to assign an argument.
4. Select the applicable argument in the **Action Arguments** pane.
5. Type the argument or select the corresponding list.
6. Select an argument from the list, if applicable.

SAVING A MACRO

✎ Discussion

After creating a macro, you must save it before you can test or run it. When you attempt to save a macro or close the Macro window without saving, a Save As dialog

box opens with the default name of **Macro#** (numbered consecutively). It is best to use a short name that clearly indicates the function of the macro. The name **Macro1**, for example, does not mean much to other users who may be working with the macro.

The name that you give the macro appears in the **Macros** object list in the Database window. All the macros associated with the database being used appear in the Database window.

- You can rename a macro. However, if you rename a macro, it may no longer run, depending on how you initiate the macro. Instead, it is a good idea to copy a macro, rename it, and then edit it in order to create macros that perform similar tasks.

- You can also save a macro by selecting the **File** menu and then selecting the **Save** command.

➥ Procedures

1. Open the desired database.
2. Create the desired macro.
3. Click the **Save** button  on the **Macro Design** toolbar.
4. Type a name for the macro.
5. Select **OK**.

USING SINGLE STEP MODE FOR TESTING

✎ Discussion

Access runs macros so quickly that you may not be able to see each action as it is performed. If you wish to confirm that the actions are being performed correctly, you can step through the macro. Single step mode is particularly useful when a macro contains numerous actions or arguments, or when a macro does not seem to perform as planned.

When you step through a macro, Access pauses before each action, allowing you to view each step. The Macro Single Step dialog box opens, displaying the macro name, condition, action name, and selected arguments. The Macro Single Step dialog box has three buttons: **Step**, **Halt**, and **Continue**. You select the **Step** button to perform

the action in the dialog box. If there are no errors, the next action appears in the dialog box. If you want to stop running the macro in this mode, you select the **Halt** button. The macro stops running and the dialog box closes. When you want to stop stepping through the macro and run the remaining steps, you select the **Continue** button. The remaining macro actions are then completed.

If an error does occur, you can make the changes in the Macro Design window. After you make the changes to a macro, you should retest it.

●	You should only use single step mode when working in the Macro Design window. When you enable single step mode, it remains activated throughout the current Access session. All macros will run in single step mode until the mode is disabled.

●	You can run a macro without testing it by selecting the **Run** menu and then selecting the **Run** command or by clicking the **Run** button on the **Macro Design** toolbar.

➡ Procedures

1. Open the desired database.
2. Open the desired macro in the Macro Design window.
3. Click the **Single Step** button  on the **Macro Design** toolbar.
4. Click the **Run** button  on the **Macro Design** toolbar.
5. Select the **Step** button as necessary to complete all the macro actions.
6. Select the **Halt** button to stop running the macro and close the dialog box.
7. Select the **Continue** button to run the remaining steps.

RUNNING A MACRO

✎ Discussion

You can run a macro from the Database window by selecting the **Macros** tab and double-clicking the desired macro name or by selecting the desired macro name and selecting the **Run** button on the Database window toolbar.

- A warning box opens if required arguments are missing or if the macro cannot run for any reason.

- You can also run a macro by right-clicking the macro name in the Database window and then selecting the **Run** command.

➡ Procedures

1. Open the desired database.
2. Select the **Macros** object list.
3. Select the macro you want to run.
4. Select the  **Run** button on the Database window toolbar.

EDITING AN EXISTING MACRO

✎ Discussion

After you have created a macro, you may decide to add or delete existing actions or change actions or action arguments. In the Macro Design window, you can cut, copy, and paste to edit a macro. You can also insert and delete rows. After you make the desired changes, you must save the macro to save the changes. It is a good idea to test the edited macro using single step mode. You must remember to disable single step mode after testing a macro.

➥ Procedures

1. Open the desired database.
2. Select the **Macros** object list.
3. Select the macro you want to edit.
4. Select the  **Design** button on the Database window toolbar.
5. Edit the macro as desired.
6. Save the changes to the macro.

RUNNING A MACRO USING THE TOOLS MENU

✎ Discussion

You can run a macro using the **Tools** menu. This option allows you to execute a macro from almost anywhere. When the Run Macro dialog box opens, you can either type the name of the macro in the text box or select the macro name from a list.

➥ Procedures

1. Open the desired database.
2. Display the **Macros** object list.
3. Select the **Tools** menu.
4. Point to the **Macro** command.
5. Select the **Run Macro** command.
6. Select the **Macro Name** list.
7. Select the macro you want to run.
8. Select **OK**.

CREATING A MACRO USING THE MACRO BUILDER

✎ Discussion

You can create a macro using the Macro Builder. The Macro Builder is used when you want to associate a macro with an existing object or control. The steps for creating a macro using this method are similar to creating a macro using the **New** button on the Database window toolbar.

Creating a macro using this method requires that you associate the macro with an event, such as **On Click** or **On Close**. When you select an **Event** property, the **Build** button appears. The **Build** button gives you access to the Macro Builder.

➥ Procedures

1. Open the desired database.
2. Open the desired form or report in **Design** view.
3. Select the desired object or control to associate the macro.
4. Click the **Properties** button  on the **Form Design** toolbar.
5. Select the **Event** tab.
6. Select the desired **Event** property.
7. Click the **Build**  button.
8. Select **Macro Builder** from the list box.
9. Select **OK**.
10. Type the macro name.
11. Select **OK**.

SECTION 5 -
USING MACROS

In this section, you will learn how to:

- Use properties
- Assign a macro to a control
- Create a command button
- Add a condition to a macro
- Create a group macro
- Create an Autoexec macro

USING PROPERTIES

✎ Discussion

Properties allow you to specify the appearance and behavior of objects in a database. Objects include tables, queries, forms, and reports, as well as controls within reports or forms.

Property sheets display the properties of a selected object. They have several tabs including **Format**, **Data**, **Event**, and **Other** that list the properties by group. The groups are the same for every property, but the items in the group change depending on the type of object selected. The **All** page displays all the properties in a single list.

The **Format** properties allow you to control the appearance of an object, such as color, font, size, and borders. These properties change automatically when you make changes to an object. The **Data** properties allow you to specify the source of the data and control items, such as default values. The **Event** properties allow you to control when an action occurs. The **Other** properties contain items that do not fit into the other three categories, such as the name of a control when used in a macro or text that appears in the status bar.

ASSIGNING A MACRO TO A CONTROL

✎ Discussion

You can associate a macro with a control on a form or report using the **Event** properties of the control. An event is an action, such as a mouse click or a change in value that can initiate a response. The macro runs automatically when the specified event involving that control occurs.

Many events involve the control having the focus. Focus means that the control can receive data from mouse clicks or keyboard actions. For example, field boxes, toggle buttons, and radio buttons can have the focus, since they can respond to data entry from the keyboard or mouse. Only one control can have the focus at a time. The **Event** properties of a text box control are listed in the following table:

Event Property	Action is initiated
Before Update	before the data in a control is updated
After Update	after the data in a control is updated
On Change	when the data in a control is updated
On Enter	when the control first receives the focus from another control on the same form

Event Property	Action is initiated
On Exit	when the control loses the focus to another control on the same form
On Got Focus	when the control gets the focus from another form or control
On Lost Focus	when the control loses the focus to another control or form
On Click	by clicking and releasing the primary mouse button on the control
On Dbl Click	by clicking and releasing the primary mouse button twice on the control
On Mouse Down	by clicking the mouse button while the mouse pointer is on a control
On Mouse Move	by moving the mouse pointer over a control
On Mouse Up	by releasing the mouse button while the mouse pointer is over the control
On Key Down	by pressing any key on the keyboard when a control has the focus or is using a **SendKeys** macro
On Key Up	by releasing a key or immediately after running the **SendKeys** macro
On Key Press	by pressing and releasing any key on the keyboard when on a control that has the focus or when using a **SendKeys** macro

By linking a command button or a control with a macro, you can make commonly used functions easier. Initiating macros this way is particularly useful because it does not require the user to know or use the Access menu structure. For example, you could associate a macro with a command button on a form to print a report. The user could then print the report by clicking the command button, without having to open the report or know the correct menu commands.

- You can also open the property sheet for a control by double-clicking the control, selecting the **View** menu and then selecting the **Properties** command, or by right-clicking the control and then selecting the **Properties** command.

- You can also create an expression using the Expression Builder by clicking the **Expression Builder** button to the right of any property on the **Event** page in the Properties dialog box, by clicking the **Build** button on the **Form Design** toolbar, by right-clicking in the desired property on the **Event** page and then selecting the **Build** command, or by pressing **[F4]** on the keyboard.

➡ Procedures

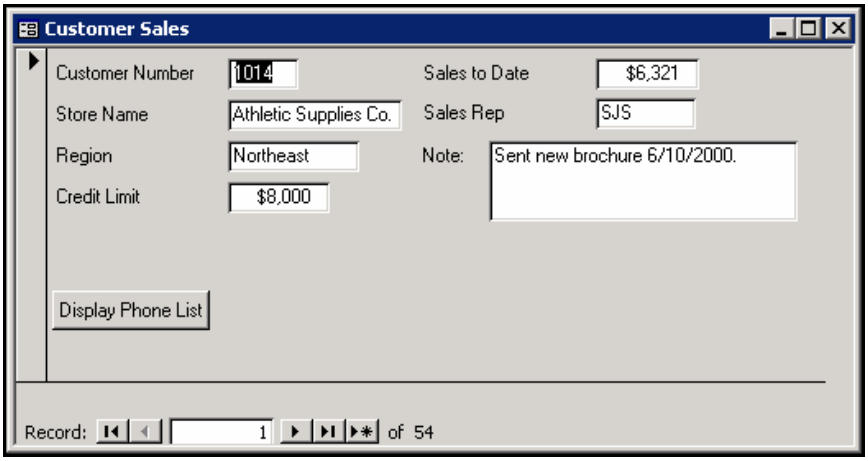
1. Open the desired database.
2. Open the desired form in **Design** view.
3. Select the desired control.
4. Click the **Properties** button  on the **Form Design** toolbar.
5. Select the **Event** tab.
6. Select the desired property.
7. Select the property list.
8. Select the desired macro.
9. Click the **Close** button to close the property sheet.

CREATING A COMMAND BUTTON

✎ Discussion

Command buttons are controls to which you can assign actions. They are often used to run macros in a form or report. You can place descriptive text or a picture on a command button so that the user can easily identify the purpose of the button.

You create a command button in **Design** view using the **Command Button** tool in the toolbox. After you create the command button on the form, you set the properties, such as the event property to initiate the action and the text or picture to appear on the button.



A command button created on a form

- To create a command button in which a macro will be attached, it is recommended to disable the **Control Wizards** button in the toolbox.

- You can use the Control Wizard to create a command button that allows you to synchronize records between two forms. As you move through records on a form, you can view related records on a second form by clicking the command button.

➡ Procedures

1. Open the desired database.
2. Open the desired form or report in **Design** view.
3. Display the toolbox.
4. Click the **Control Wizards** tool  in the toolbox to disable it, as necessary.
5. Click the **Command Button** tool  in the toolbox.
6. Click in the desired location in the form or report where you want to create the command button.
7. Click the **Properties** button  on the **Form Design** or **Report Design** toolbar.

8. Select the **Event** tab.
9. Select the desired property.
10. Select the property list.
11. Select the desired macro.
12. Select the **Format** tab.
13. Select the text in the **Caption** property.
14. Type the text you want to appear on the button.
15. Click the **Close** button to close the property sheet.

ADDING A CONDITION TO A MACRO

✎ Discussion

You can add a condition argument to a macro. An argument acts very much like a filter. Just as a filter displays only the records that meet the condition, the argument only performs the macro when the condition is met.

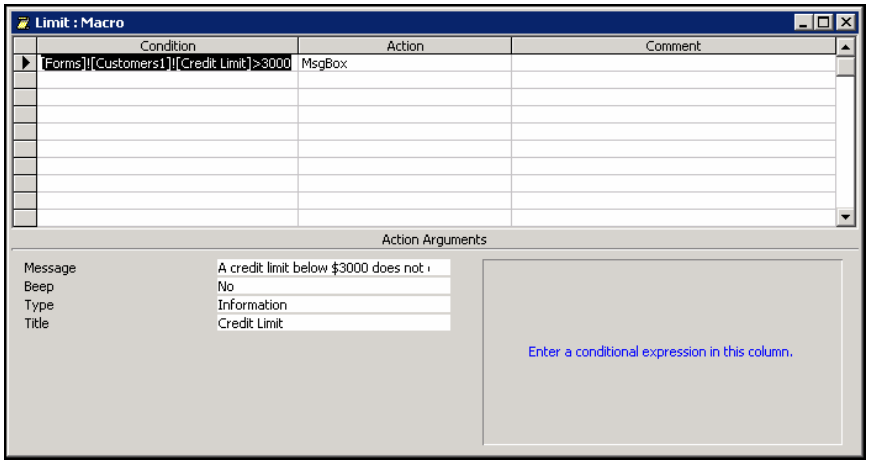
Before you can add a condition to a macro, you must display the **Condition** column in the Macro Design window. The **Condition** column appears to the left of the **Action** column in the upper pane of the Macro Design window. You can type a condition or use the Expression Builder dialog box to create the expression.

If you type an expression, there are certain rules you must follow when referring to controls in tables, queries, forms, and reports. All references must be separated with an exclamation point. Additional rules for the database objects and controls are listed in the following table:

Controls	Rules
Controls in tables	Enclose the name of the table and the name of the control in square brackets and separate them with an exclamation point. For example, **[Orders]![Customer ID]** refers to the **Customer ID** field in the **Orders** table.
Controls in queries	Enclose the name of the query and the name of the control in square brackets and separate them with an exclamation point. For example, **[Order Items]![Item Number]** refers to the **Item Number** field in the **Order Items** query.

Controls	Rules
Controls in forms	Enclose the name of the form and the name of the control in square brackets and separate them with an exclamation point. Indicate that you are referring to a form by beginning the statement with the word **Forms**. For example, **Forms![Customers]![Customer ID]** refers to the **Customer ID** field in the **Customers** form.
Controls in reports	Enclose the name of the report and the name of the control in square brackets and separate them with an exclamation point. Indicate that you are referring to a report by beginning the statement with the word **Reports**. For example, **Reports![Customer Sales]![Contact Name]** refers to the **Contact Name** field in the **Customer Sales** report.

Since you enter the condition in the **Condition** column in the Macro Design window, you do not need to include the word **IF** in the statement. Access assumes the statement is a condition. If the condition is true, Access performs the corresponding action in that row. If the condition is false, Access does not perform the action.



Adding a condition to a macro

- You can also create an expression using the Expression Builder when you display the **Condition** column by right-clicking in the **Condition** column and then selecting the **Build** command, by right-clicking in the **Action Arguments** pane and then selecting the **Build** command, or by clicking the **Build** button on the **Macro Design** toolbar.

- If the condition is true, you can have Access perform more than one action by entering an ellipsis (...) in the **Condition** column in the Macro Design window. Access performs the action on the same row as the condition and all rows thereafter that contain an ellipsis.

- You can also display the **Conditions** column by selecting the **View** menu and then selecting the **Conditions** command.

➥ Procedures

1. Open the desired database.
2. Open the desired macro in the Macro Design window.
3. Click the **Conditions** button  on the **Macro Design** toolbar.
4. Select the field in the **Condition** column next to the applicable action.
5. Type the condition in the field.

CREATING A GROUP MACRO

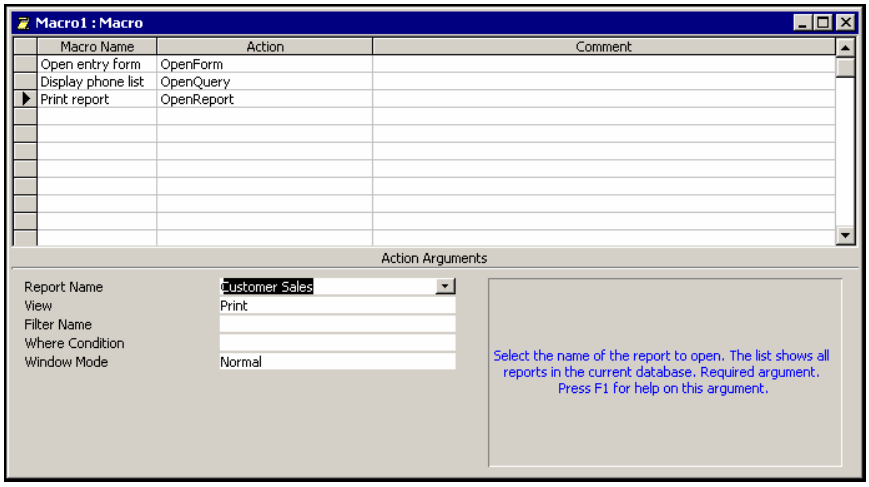
✎ Discussion

You can place multiple macros on the same macro sheet. This option, called grouping macros, has several advantages. The most obvious advantage is that it reduces the number of macros that appear in the window. More importantly, it allows you to organize your macros by grouping related macros together onto a single macro sheet. For example, if you created several macros that are related to one form, you can group all of them together on one macro sheet.

Before you can create a group macro, you need to display the **Macro Name** column in the Macro Design window. The **Macro Name** column appears to the left of the **Action** or **Condition** column, if it is displayed. You can type the macro name directly into the appropriate field in the **Macro Name** column.

Like any other macro, a group macro must be saved with a name. You should make sure that the group macro name is descriptive enough to indicate its contents. This helps other users know what macros are contained in the group macro. For example, if all the macros relate to the **Orders** form, you could name the group macro **Order Form Macros**.

When macros are grouped, you can still refer to individual macros. To refer to an individual macro in a group, you can use its group and macro name in the format **Groupname.Macroname**.



Creating a group macro

- If you have already created macros, you can cut and paste them into a new group macro sheet or edit an existing macro sheet.

- You can quickly create a group macro by dragging the desired object from the Database window to the **Action** column in the Macro Design window. In doing this, the **Action** column and **Form Name** argument field are automatically filled in; you will need to type a name for the macro, however.

➥ Procedures

1. Open the desired database.
2. Display the **Macros** object list.
3. Open the Macro Design window.
4. Click the **Macro Names** button  on the **Macro Design** toolbar.
5. Select the first blank field in the **Macro Name** column.
6. Type a name for the macro.

7. Select the corresponding field in the **Action** column next to the macro name.
8. Select the **Action** list.
9. Select the desired action.
10. Select the applicable argument in the **Action Arguments** pane.
11. Select the argument list.
12. Select the desired argument.
13. Continue to add macros to the group macro as desired.

CREATING AN AUTOEXEC MACRO

✎ Discussion

While you can select a command to run a macro, you can also have a macro that runs automatically when a database is opened. This type of macro is called an Autoexec macro because it is automatically executed when the database is opened.

You can create an Autoexec macro to automate or simplify the database for other users. For example, assume that you want someone in your office to add customer orders on a specific form. To ensure that they use the correct form, you could create an Autoexec macro that opens the form for them.

You can create an Autoexec macro in two ways. You can rename an existing macro **Autoexec**, or you can create a new macro and save it with the name **Autoexec**. Since the Autoexec macro is stored with the database, each database can have its own Autoexec macro. However, you can only have one Autoexec macro per database.

- You can bypass the Autoexec macro by holding the **[Shift]** key while you open a database.

- You can also rename a macro by selecting the desired macro in the Database window, selecting the **View** menu, and then selecting the **Rename** command.

➡ Procedures

1. Open the desired database.
2. Display the **Macros** object list.
3. Right-click the macro name you want to rename.
4. Select the **Rename** command.
5. Type **Autoexec**.
6. Press **[Enter]**.

SECTION 6 -
CREATING CUSTOM TOOLBARS

In this section, you will learn how to:

- Create a custom toolbar
- Add a built-in menu item
- Create a custom menu item
- Add a command to a menu item
- Add custom commands
- Dock a floating toolbar
- Link a toolbar to a report
- Add a separator bar
- Create a shortcut menu
- Link a custom shortcut menu
- Add buttons to a custom toolbar
- Change a button image
- Delete a custom toolbar

CREATING A CUSTOM TOOLBAR

✎ Discussion

When you create a custom application, you might also want to create custom menus. Custom menus allow you to limit the actions a user can perform. For example, you can remove certain commands from the **File** menu, such as **New** and **Rename** to prevent users from modifying the application database. You can also add a custom command that initiates a macro.

Menus and command buttons reside on toolbars. Usually a toolbar contains only menus or only buttons. However, you can display both on the same toolbar.

Once you have created a custom toolbar, you can link it to a form or a report, meaning that it will automatically appear when you open or run the form or report.

- A new toolbar appears as a floating toolbar.
- You can also open the Customize dialog box by selecting the **Tools** menu and then selecting the **Customize** command.
- The toolbar will automatically resize when you add custom menus and buttons.

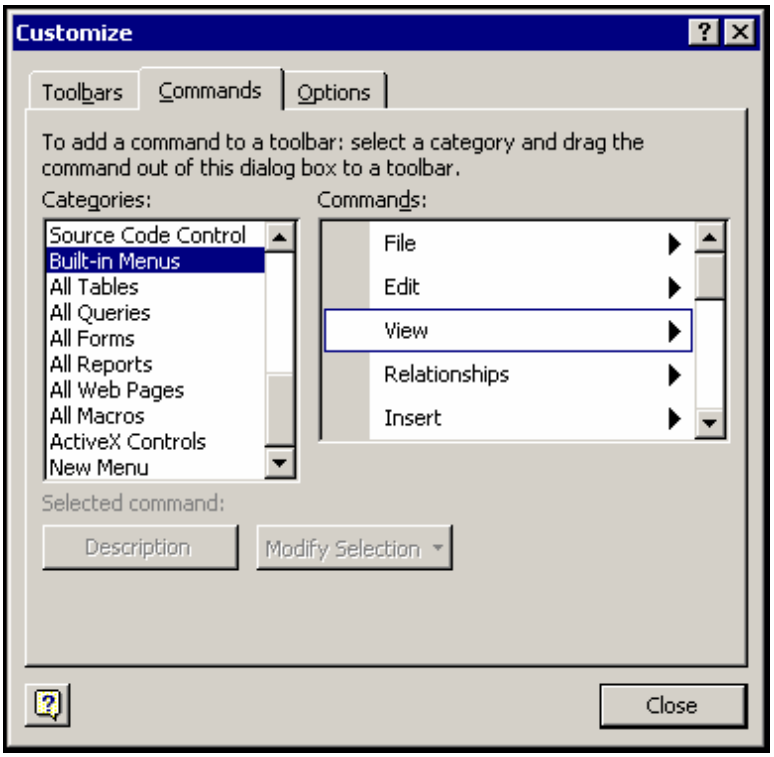
➡ Procedures

1. Open the desired database.
2. Select the **View** menu.
3. Point to the **Toolbars** command.
4. Select the **Customize** command.
5. Select the **Toolbars** tab.
6. Select the **New** button.
7. Type a descriptive name for the new toolbar.
8. Select **OK**.
9. Select **Close**.

ADDING A BUILT-IN MENU ITEM

✎ Discussion

When the Customize dialog box opens, any toolbar can be displayed and edited. Menu items are listed by category, each of which contains a group of related menu commands. For instance, the **Edit** category, when selected, displays editing commands such as **Copy** and **Paste**. You can quickly add or remove entire menu items or individual menu commands to and from existing toolbars in Access. For example, you may wish to remove the **File** menu to prevent users from performing any file operations. The Customize dialog box includes built-in menus you can add to a toolbar. These built-in menu items already include a group of commands.



Adding a built-in menu item

- When adding Built-in Menu items to a custom toolbar, you cannot add items that are not relative to the object window.

➥ Procedures

1. Open the desired database.
2. Display the desired toolbar.
3. Select the **View** menu.
4. Point to the **Toolbars** command.
5. Select the **Customize** command.
6. Select the **Toolbars** tab.
7. Click the option to the left of the toolbar you want to edit.
8. Select the **Commands** tab.
9. Select **Built-in Menus** from the **Categories** list box.
10. Select the desired built-in menu from the **Commands** list box.
11. Drag the built-in menu to the desired toolbar.
12. Close the Customize dialog box.

CREATING A CUSTOM MENU ITEM

✎ Discussion

A quick way to develop a custom toolbar is by adding built-in menus. Built-in menus are menu items that already contain a group of commands. However, you may want to create your own menu item to which you can assign a group of commands. Access allows you to create a new menu item to which you can assign a name and add commands that appear in a list when the menu is selected.

➥ Procedures

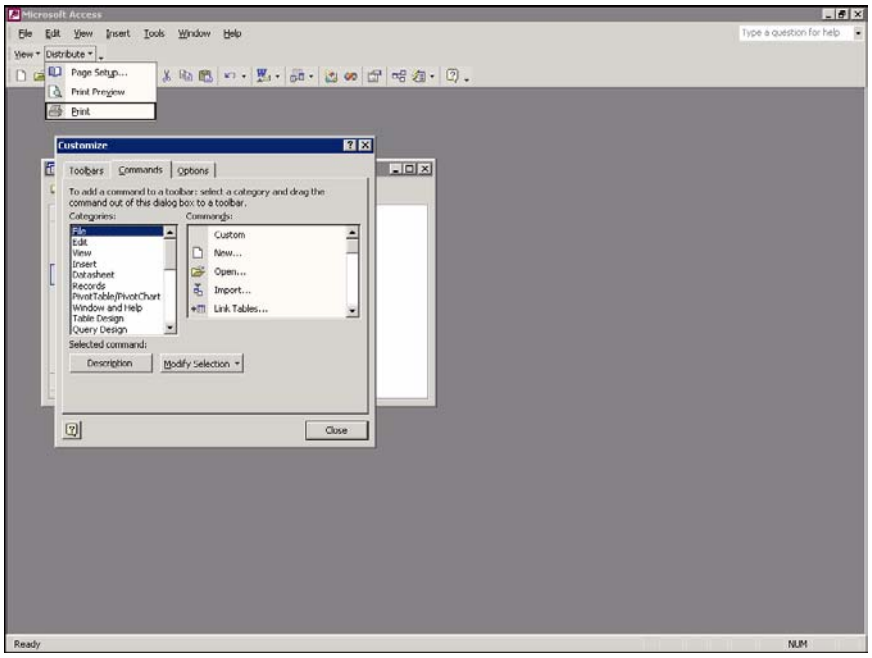
1. Open the desired database.
2. Open the Customize dialog box.
3. Select the **Commands** tab.
4. Select **New Menu** from the **Categories** list box.
5. Select **New Menu** from the **Commands** list box.

6. Drag the **New Menu** menu to the desired toolbar.
7. Select the **Modify Selection** button in the Customize dialog box.
8. Drag to select the text in the **Name** box.
9. Type a descriptive name for the new menu item.
10. Press **[Enter]**.
11. Close the Customize dialog box.

ADDING A COMMAND TO A MENU ITEM

✎ Discussion

You can add commands to a menu item when the Customize dialog box is open. Related commands are grouped in categories. For instance, the **File** category contains file handling commands, such as **Open Database**, **Save**, and **Print**. You may want to add commonly used commands to a menu item if they are not already there. When you add a command, a vertical or horizontal bar appears, indicating where the command will be added.



Commands added to a menu item

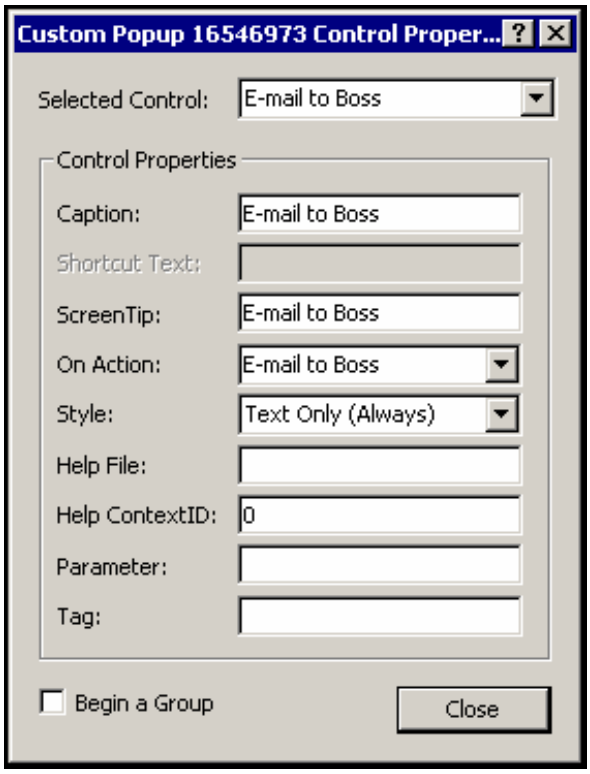
➡ Procedures

1. Open the desired database.
2. Open the Customize dialog box.
3. Select the **Commands** tab.
4. Select the desired category from the **Categories** list box.
5. Select the desired command from the **Commands** list box.
6. Drag the command to the menu name on the desired toolbar and then place it in the desired location.
7. Close the Customize dialog box.

ADDING CUSTOM COMMANDS

✎ Discussion

Most of the commands available in the Customize dialog box already have an action associated with them. For example, the **Print** command, when selected, performs the action of printing the active object, such as a form or report. However, the **Custom** command, which is included in the selection of available commands, has no action associated with it when added to a menu item. You can assign a macro to this command so that when it is selected, it performs the actions of that macro. You can also rename the **Custom** command to provide a description of the actions that the command performs.



A custom command added to a menu item

➡ Procedures

1. Open the desired database.
2. Open the Customize dialog box.
3. Select the **Commands** tab.
4. Select **File** from the **Categories** list box.
5. Select **Custom** from the **Commands** list box.
6. Drag the **Custom** command to the menu name on the desired toolbar and then place it in the desired location.
7. Select the **Modify Selection** button in the Customize dialog box.
8. Drag to select the text in the **Name** box on the **Modify Selection** menu.
9. Type a descriptive name for the command.
10. Select the **Properties** command on the **Modify Selection** menu.
11. Select the **On Action** list.

12. Select the macro you want to assign to the command.
13. Close the Customize dialog box.

DOCKING A FLOATING TOOLBAR

✎ Discussion

When you create a custom toolbar, it appears as a floating toolbar by default, meaning that it appears as a small window that can be easily moved and closed. Toolbars can be docked horizontally along the top or bottom of the Access window, or vertically along either side of the window. This flexibility allow you to place the toolbar in the location you prefer.

➥ Procedures

1. Open the desired database.
2. Display the floating toolbar.
3. Drag the toolbar to the desired edge of the Access window.

LINKING A TOOLBAR TO A REPORT

✎ Discussion

After you have created a custom toolbar, you can link it to a form or report. When the form or report opens, the custom toolbar appears along with any default toolbars.

➥ Procedures

1. Open the desired database.
2. Display the **Forms** or **Reports** object list.
3. Select the desired form or report.
4. Select the  **Design** button on the Database window toolbar.

5. Click the **Properties** button  on the **Report Design** or **Form Design** toolbar.
6. Select the **Other** tab.
7. Select the **Toolbar** property.
8. Select the **Toolbar** list.
9. Select the desired toolbar.

ADDING A SEPARATOR BAR

✎ Discussion

You can create separator bars on menus to group related commands. For example, the **File** menu in Access has several separator bars that distinguish groups of file operations, such as opening, closing, saving, and printing. You can add separator bars to menus you have created to group related commands.

➥ Procedures

1. Open the desired database.
2. Display the toolbar to which you want to add a separator bar.
3. Open the Customize dialog box.
4. Select the **Commands** tab.
5. Display the menu on the toolbar to which you want to add a separator bar.
6. Select the command before which you want to insert a separator bar.
7. Select **Modify Selection** in the Customize dialog box.
8. Select the **Begin a Group** command.
9. Close the Customize dialog box.

CREATING A SHORTCUT MENU

✎ Discussion

Shortcut menus appear when you right-click in an application. You can use shortcut menus to select commonly used commands rather than having to locate the commands in a menu. You can link a shortcut menu to a form or report, or you can have it appear globally.

- To have a custom shortcut menu bar appear globally, you need to change the option in the **Shortcut Menu Bar** text box in the Startup dialog box from **<default>** to the name of the custom shortcut menu. Changes made in the Startup dialog box do not take effect until the next time the database is opened. Custom shortcut menus linked to forms and reports replace the global custom shortcut menu.

➥ Procedures

1. Open the desired database.
2. Open the Customize dialog box.
3. Select the **Toolbars** tab.
4. Select the **New** button.
5. Type a descriptive name for the shortcut menu.
6. Select **OK**.
7. Select the **Properties** button.
8. Select the **Type** list.
9. Select **Popup**.
10. Select **OK**.
11. Select **Close**.
12. Double-click **Shortcut Menus** in the **Toolbars** list box.
13. Select the **Custom** menu on the **Shortcut Menus** toolbar.
14. Select the shortcut menu to which you want to add commands.
15. Select the **Commands** tab in the Customize dialog box.

16. Select the desired category in the **Categories** list box.
17. Select the desired command in the **Commands** list box.
18. Drag the command to the desired location on the custom shortcut menu on the **Shortcut Menus** toolbar.
19. Close the Customize dialog box.

LINKING A CUSTOM SHORTCUT MENU

✎ Discussion

After you have created a custom shortcut menu, you can link it to a form or report. If the form is open in **Form** view or the report is open in **Print Preview**, the shortcut menu appears when you right-click in the form or report.

A custom shortcut menu linked to a form

➡ Procedures

1. Open the desired database.
2. Display the **Forms** or **Reports** object list.
3. Select the desired form or report.
4. Select the **Design** button on the Database window toolbar.
5. Click the **Properties** button on the **Form Design** or **Report Design** toolbar.

6. Select the **Other** tab.
7. Select the **Shortcut Menu Bar** property.
8. Select the **Shortcut Menu Bar** list.
9. Select the desired shortcut menu.

ADDING BUTTONS TO A CUSTOM TOOLBAR

✎ Discussion

You can add command buttons to a custom toolbar. Command buttons, when clicked, perform an action, such as saving a file or opening a database object. Generally, the action is depicted in a graphic on the button. For example, the **Save** button has a graphic of a floppy disk on it. The Customize dialog box allows you to select command buttons from several categories. You can create a toolbar that contains only command buttons, or you can mix menu items with command buttons. The **Commands** list in the Customize dialog box displays the text for the commands as well as the corresponding buttons.

- By default, when you add a command to a toolbar, only the button appears. However, you can display the text for the command as well by selecting the **Modify Selection** menu and then selecting the **Image and Text** command. When you add a command from the **All Macros** category, on the other hand, the button and command text appear by default. You can disable the display of the command text by selecting the **Modify Selection** menu and then deselecting the **Image and Text** command.

- To remove a button from a toolbar, you can drag it off the toolbar when the Customize dialog box is open.

- To rearrange buttons on a toolbar, you drag them to the desired locations when the Customize dialog box is open.

➥ Procedures

1. Open the desired database.
2. Open the Customize dialog box.

3. Display the desired toolbar.
4. Select the **Commands** tab.
5. Select the desired category from the **Categories** list box.
6. Select the desired command in the **Commands** list box.
7. Drag the command to the desired toolbar.
8. Close the Customize dialog box.

CHANGING A BUTTON IMAGE

✎ Discussion

When you add a button to a custom toolbar, Access displays the default button. When you add an object to a custom toolbar, Access displays the default button for that type of object. You can change the face of any button on any toolbar. Access includes a button editor so that you can modify an existing button image or create your own.

- If you do not want to use the available button images, you can create your own graphic by selecting the **Modify Selection** menu and then selecting the **Edit Button Image** command.

➥ Procedures

1. Open the desired database.
2. Display the desired toolbar.
3. Open the Customize dialog box.
4. Select the command button you want to change.
5. Select **Modify Selection** in the Customize dialog box.
6. Point to the **Change Button Image** command.
7. Select the desired button image.
8. Close the Customize dialog box.

DELETING A CUSTOM TOOLBAR

✎ Discussion

You can delete a custom toolbar when you no longer need it. Deleting toolbars is a good way to keep your desktop and the Customize dialog box free of clutter.

- You can delete a custom shortcut menu by selecting the **Shortcut Menus** toolbar on the **Toolbars** page in the Customize dialog box and then selecting **Properties**. In the Toolbar Properties dialog box, you can select the shortcut menu you want to delete in the **Selected Toolbar** list and then change the toolbar type to **Menu Bar** in the **Type** list. The shortcut menu will then appear in the **Toolbars** list box in the Customize dialog box and you can delete it just as you would any other toolbar.

➡ Procedures

1. Open the desired database.
2. Open the Customize dialog box.
3. Select the **Toolbars** tab.
4. Select the toolbar you want to delete in the **Toolbars** list box.
5. Select **Delete**.
6. Select **OK**.
7. Close the Customize dialog box.

INDEX

For Training and Material on Other Topics,
please visit the Education and Training web site at:
webster/saa/services/training
or
Call the Education and Training registration number at:
(202) 224-7628